BASIC TECHNOLOGIES FOR WEB ENGINEERING
BUILDING A WEB APPLICATION THAT ALLOWS THE NAVIGATION AND SEARCH OF MAIL ARCHIVES

Nayef Fayez Roqaya - Ahmad Alzeitoun

B-IT center, University of Bonn, Master Computer Science SS17

ABSTRACT

The main objective is to implement a server-side component with the node.js platform, which allows the development of server-side JavaScript applications. We create a Web application (HTML5-based GUI) that allows the navigation and search of mail archives.

***Index Terms*—** JavaScript, Node.js, data-intensive, non-blocking, asynchronous, simplifies

1. INTRODUCTION

There are may programming languages that are used in web development field. One of the most important language nowaday is Node.js. Basically, It is a platform that built on Chrome's JavaScript runtime for building flexible, scalable and fast network applications. On the one hand, Node.js relays on many concepts such as event-driven, non-blocking model that makes it perfect and efficient for data-intensive real-time applications which run through distributed devices. On the other hand, Node.js is used for developing server-side and distributed applications additionally, it is an open source. It is very important to know the answer of very important question about Node.js about how this language was written [4]. Node.js applications are written in JavaScript, and can be run within the Node.js runtime different operating systems. Oversimplification, Node.js also provide us with library of various JavaScript modules which contribute in simplify the development of web applications depending on Node.js to a great extent. From an abstract point of view, alot of features put node.js in the first class when we talk about choosing the best choice becuase it has many usfullness such as such as asynchronous and event driven, very fast, single threaded but highly scalable and no buffering [3]

2. PROBLEM DEFINITION

In this report, we illustrate how we can implement Web application that allows the navigation and search of mail archives depending on implementing a set of RESTful Web Services, support libraries are mailbox parsers such as MailParser and/or node-mbox.

3. APPROACH

Our Solution approach depends on npm Enterprise where it helps large teams share, discover, and re-use code behind the corporate firewall so they can build amazing things. We used node-mbox for reading the (File.mbox) then use MailParser to extract and mapping the emails that are existed inside the file. Additionally, we support our approach with many modules that will be explained in details in the next section. Figure1 illustrates the main steps in our solution approach and how it starts and where it completes. The first step is reading the mbox file as an input by node-mbox then parsing the emails by mailparser. After these steps, we can apply many different modules that represent the functionalities in our systems.

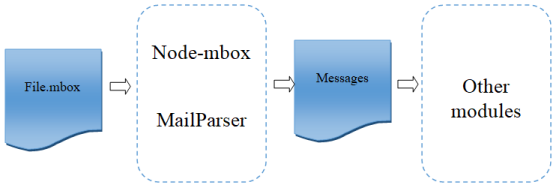

Fig. 1. Solution approach

4. IMPLEMENTATION

In this section, we display the modules that are implemented in our system. Our modules are divided into two types. The first is reading the mbox and getting all emails inside this file and the second type includes many different modules that involve many functions such as uploading, mapping, navigating and searching. Uploading function basically upload the mbox file as an input and this module is the core because we can not apply any other module before this important step. Mapping module depends on using mailparsing that extract the details of the messages from the mbox file. Navigating modules give the ability to navigate between the message by next and previous press button. Finally, searching module gives the ability to search a specific message depending on a specific information. Figure2 display the use case diagram that include the

basic modules and functions in our system. regarding the mail parser, there are two separate modes, a lower level MailParser class and simpleParser function. The latter is simpler to use but is less resource efficient as it buffers attachment contents in memory. SimpleParser is an easy way for parsing the email wgere we need only the message source then it will return the email structure. Mail object includes many properties such as headers, subject, from, to, CC, BCC, date, attachments...etc. Regarding the functions that are already implemented, we implemented function for uploading the mbox file, function for search message in the inbox, function for mapping that parses the data from the message and functions for navigating. When we talk about interoperability between computer systems on the Internet, we can know that the RESTful web services are built to work best on the Web. REST-compliant Web services allow requesting systems to access and manipulate textual representations of Web resources using a uniform and predefined set of stateless operations. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol [2].

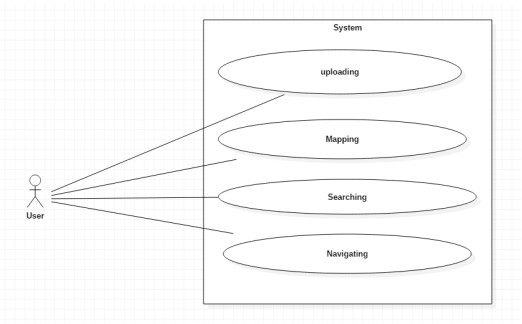

Fig. 2. Use case diagram

Figure3illustrates the simple and effecient GUI for our system. This GUI implemented depending on HTM5 where the HML code and styling code were implemented and saved in embedded java script file ”ejs”. We concentrated on simplification in our GUI design to be easy use, but at the same time we considered the efficiency and offer many functionality by this GUI [1]

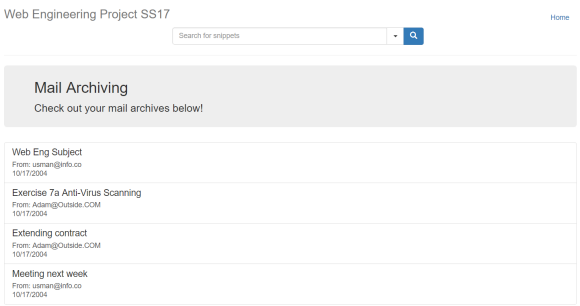

Fig. 3. GUI of the system

5. EVALUATION

This stage is one of the most important stages in this report because it reflects the quality of our system and abilities of our functions that are already implemented. For tests the system, we imported our gmail mail boxes as a file include manay emails with many different information additional to consider existing the attachment or not. The test completed successfully and gave us all the results that are expected. The figure

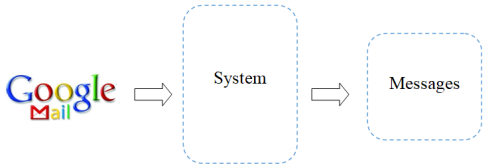

Fig. 4. Evaluation and test process

6. SUMMARY

Summing up, Node.js being single-threaded means that one does not need to care about the problems or shared mutable state and synchronising between threads. Also, npm and its structure make diffecult to find trustable packages. Using Node.js in implementing this type of the system gives the system many extra properties such as distributed, flexibility and easy use.

7. REFERENCES

[1] Node mailer Proudly Sponsored by MoonMail. Mail parser, 2017. https://nodemailer.com/about/.

[2] Oracle. The java ee 6 tutorial, 2017. RESTful Web Services, http://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html.

[3] tutorialspoint. Node.js, 2017. https://www.tutorialspoint.com/.

[4] Dr.Carlos A Velasco. Basic technologies for web engineering, 2017. B-IT center, https://fit-bscw.fit.fraunhofer.de.